

Papers that [reference](#) this paper.

Reconstructing Irregularly Sampled Images by Neural Networks

Albert J. Ahumada, Jr. and John I. Yellott, Jr.

NASA Ames Research Center, MS-262-2, Moffett Field CA 94035

and

Cognitive Sciences, University of California, Irvine CA 92717

Abstract

Neural-network-like models of receptor position learning and interpolation function learning are being developed as models of how the human nervous system might handle the problems of keeping track of the receptor positions and interpolating the image between receptors. These models may also be of interest to designers of image processing systems desiring the advantages of a retina-like image sampling array.

Introduction

The human visual system differs from most artificial image acquisition systems in that the photoreceptors are not regularly arrayed. The variable density of the retina is well known, but the irregularity of array has only recently received quantitative attention. The cones, which subserve vision at normal light levels, can be found in small neatly packed hexagonal arrays in the central fovea, but these small arrays are rarely larger than 0.1 degree and they are irregularly oriented. The night-vision-serving rods are absent from the center of the retina, but are more numerous than cones at 1 degree of eccentricity and completely isolate the cones from each other by 2 degrees. At this point the cone array is quite disordered (1,2).

Designers of artificial systems may be attracted to the advantages of continuously varying resolution: a wide field of view and high peak resolution with relatively few photoreceptors. They may also desire the advantages of irregular sampling: graceful image degradation in the "Nyquist" spatial frequency region and unstructured aliasing (3,4). The problem then arises of how to interface a retina-like array with the rest of an image processing system which is designed around rectangularly arrayed pixels.

The human visual system appears to have solved a similar problem in that the variable grain of the retina is not apparent to us and the apparent fineness of visual detail remains high over several degrees of visual angle. We hypothesize that the visual system may actually interpolate the image between retinal samples as suggested by Barlow (5), so that various visual subsystems can sample the image according to their own needs. Our models for how the visual system might provide for this interpolated image are mechanisms for interfacing irregularly sampled images to regular arrays.

The models will be described in two separate sections. The first section is concerned with the problem of how the positions in the sampling array could be known in detail by higher centers. When the connecting tracts develop, the general arrangement of the cone array is preserved, but it is difficult to see how the exact positions could be known without some calibration process. The second section is concerned with the construction of interpolation functions for reconstructing the sampled image between the irregularly positioned samples.

Section 1: Learning Maps of Receptor Positions

We have developed a self-organizing learning procedure which computes the output positions of connections mapping input units into output units. The procedure iteratively tries to minimize the difference between measures of the input distance and the output distance. Transformations of the input distance can allow the procedure to compute transformed maps with variable magnification. The activation of the process is assumed to be spontaneous activity during development rather than visual input as in the position

calibration scheme of Maloney (5,6). When there is no exact solution, the mapping will only stabilize if the learning rate goes to zero with time. Like the models of Kohonen (7) and Ritter and Schulten (8), this model assumes that there is a pre-existing mapping from input space to output space and that the learning process moves the output positions to "improve" the mapping.

We assume an array of n receptors having positions in this first layer specified by cartesian coordinates,

$$P(i) = (x(i), y(i)), i = 1, \dots, n. (1)$$

The distance from the i th to the j th receptor will be denoted by

$$d(i,j) = [(x(i)-x(j))^2 + (y(i)-y(j))^2]^{0.5}. (2)$$

Each receptor also has a connection to a position at the second, higher level denoted by

$$P'(i) = (x'(i), y'(i)), i = 1, \dots, n. (3)$$

The adjustment process begins with a single receptor becoming spontaneously active, say unit i . This activity spreads along two distinct paths. In one path the activity spreads first laterally in the receptor layer to all inactive units according to a lateral spread function,

$$s(d(i,j)) = s(i,j). (4)$$

This activity level is then transmitted up the line of the non-active unit j to the higher level. The second path for the spontaneous activity proceeds up the line of the active unit where it spreads laterally to the endpoints of the other units in the higher level according to another spread function,

$$s'(d'(i,j)) = s'(i,j). (5)$$

The functions $s()$ and $s'()$ are assumed to be strictly monotonic functions of distance.

A comparison process computes the difference between the two activity levels and forms an error signal,

$$e(i,j) = c(s'(i,j) - s(i,j)), (6)$$

where $c()$ is a sign-preserving transformation if $s()$ and $s'()$ are monotonically decreasing in d (the physiologically plausible case) or sign-inverting if they are monotonically increasing. The gain of $c()$ must be low enough to keep the process stable. Finally, the higher level point position of the inactive point is moved along the line between the two points by the rule,

$$P'(j) \leftarrow P'(j) + e(i,j)(P'(j) - P'(i)). (7)$$

The arrow (\leftarrow) indicates replacement. The high level point position of the inactive point is thus moved in a direction which will tend to correct the distance error.

Figure 1 shows an example simulation of the process. Each point representing a position has been connected by lines to its near neighbors in the input array. The receptor array coordinates were normalized to have an average spacing in the neighborhood of unity. The spread functions were identical Cauchy distributions:

$$s(d) = s'(d) = 1/(1+d^2). (8)$$

The comparison function was the identity function,

$$c(d) = d. (9)$$

The selection of the active receptor was done at random without replacement.

The receptor input positions are shown in Figure 1a. They are central foveal cone positions measured by Hirsch and Hylton (9). The initial positions of the points in the higher layer are shown in Figure 1b. They were chosen by adding uniform random variables on the interval (0,4) to the receptor array x and y coordinates. We call the selection of one point for activation and the adjustment of the other $n-1$ positions an update and n updates a cycle. Figure 1c, 1d, 1e, 1f show the output positions after 1, 2, 6, and 10 cycles, respectively. After about a dozen cycles the process converged to a solution with negligible error in the distances. The orientation and offset of the output array is not constrained by the process. Both the size and the offsets in Figure 1 were normalized for display convenience. Similar results have been obtained with arrays whose spacing variability is typical of that in the fovea and in the periphery, with both gaussian and cauchy spread functions, and with both random and sequential activation of receptors.

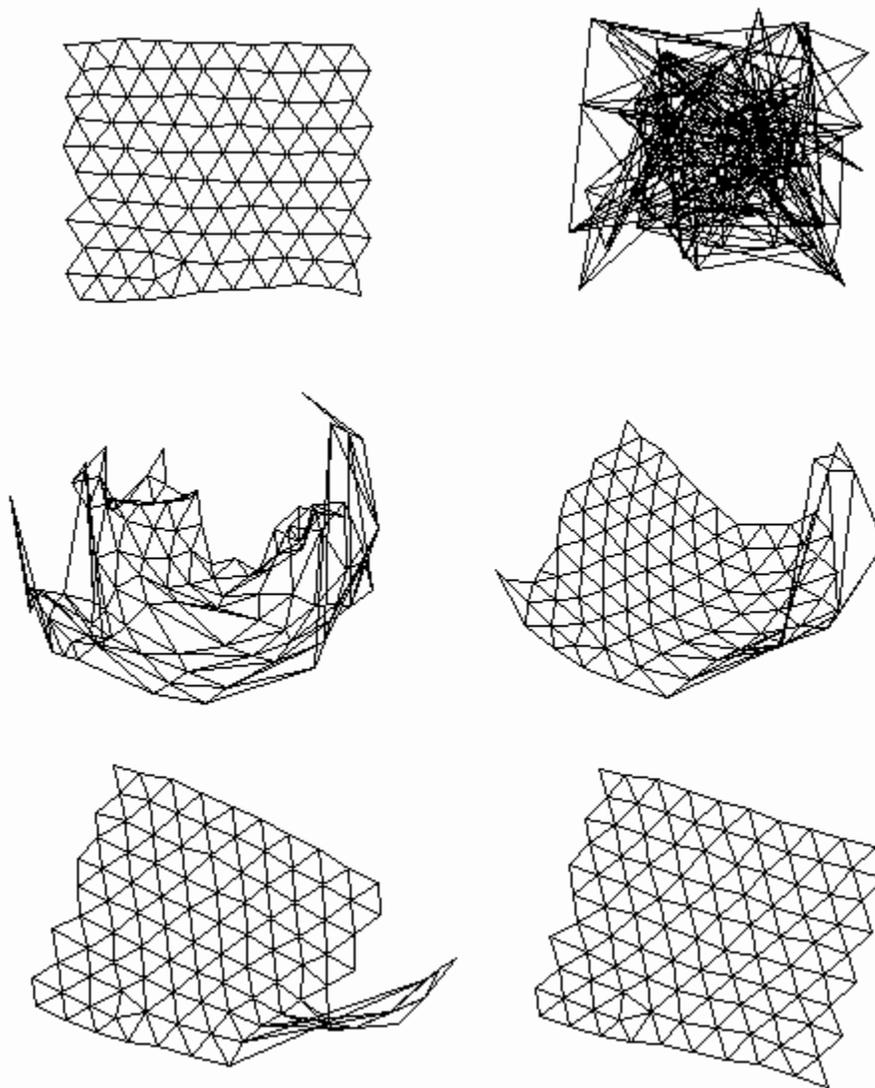


Figure 1. An example of position learning using the foveal cone position data of Hirsch and Hylton (9). a - upper left -) The original configuration. b - upper right -) The randomized positions. c - middle left -) The configuration after 1 cycle (102 updates, each comprising 101 adjustments). d - middle right -) 2 cycles. e - lower left -) 6 cycles. f - lower right -) 10 cycles.

Section 2: Interpolating Images between Receptor Positions

A system with regular sampling at the photoreceptor level can reconstruct an accurate estimate of the original image by low-pass filtering the sampled image with a space-invariant filter (10). Quality

reconstruction of images which are irregularly sampled demands space-variant filters. If the reconstruction is to be an interpolation (correct at the sample positions), the impulse response at each sample point must have zero response at all of the other sample points. The best known method for reconstructing irregularly sampled images was proposed by Yen (11). It correctly reconstructs band-limited images sampled at the Nyquist rate. The basis of the method is the definition of a regular lattice with the same number of samples over the image. First a transformation is constructed which computes what the sample values at the regular points would be given the values at the irregular points, then the standard sinc interpolation is performed between the regular sample points.

Let $V(i)$ be the irregularly sampled values, $R(j)$ be the rectangular array values, and $W(i,j)$ represent the linear transformation from V to R , so that

$$R(j) = V(i)W(i,j). \quad (10)$$

The transformation $W(,)$ turns out to be the inverse of the matrix $S(,)$ of "sinc distances" between the regular points and the irregular points,

$$S(j,i) = \text{sinc}[k(x(i)-x(j))]\text{sinc}[k(y(i)-y(j))]. \quad (11)$$

Figure 2a diagrams a unit which multiplies inputs by weights and then sums them, a type of artificial neuron which implements the operation of Equation 10. Although biological neurons are nonlinear in their output, visual system units at the cortical level have been found which behave quite linearly. Possibly the output nonlinearities in the outputs of earlier layers are compensated for in a manner similar to methods used by electronic engineers: complementary or push-pull circuitry and feedback. We consider to have constructed a neural network implementation of a system if it can be constructed of these units or other units (such as comparators or multipliers) which can easily be constructed of these units. Figure 3a shows a neural network implementation of Yen interpolation. The response to a single unit at the irregularly-spaced input layer is a linear combination of regularly-spaced sinc functions which combine to give zero response at all the other irregularly-spaced positions.

Although this procedure can be used to construct interpolation functions for irregular sampling arrays in general, the arbitrary selection of the corresponding regular array is troublesome and it is difficult to see how such a procedure could be implemented in the visual system. The method was originally intended for regular arrays with small amounts of position jitter. In this case the corresponding regular array is obvious and the matrix $S(j,i)$ is easy to invert because the small coefficients play a small role. In the case of variable density sampling, coefficients for units i and j very far apart are crucial to the inversion of the matrix, leading to serious computational problems.

Another scheme for interpolation between irregular samples has been studied by Chen and Allenbach (12). A network implementation of this method is shown in Figure 3b. Again a network transforms the inputs into outputs according to Equation 10, but now these outputs are in the same position as the inputs. Again the transformed inputs are used to adjust the gain of sinc functions centered at these values, so Equation 11 ensures that the method generates an interpolation.

Using the input positions as output positions obviates the need to define a corresponding regular array and in case of variable density, the calculation of the inverse of the $S(j,i)$ matrix is generally easier, since the inverse weights are principally controlled by nearby positions. We suffer a loss for this improvement: it is no longer as easy to characterize the set of images for which the interpolation is exact. While the Yen interpolation correctly interpolates any image band-limited to the Nyquist frequency, the Chen-Allenbach method only correctly interpolates images which are linear combinations of sinc functions centered at the sample points and projects other images into that space.

There is no need to use sinc functions or even space-invariant filters to perform interpolation. Any smoothly disappearing functions, $S(j,x,y)$, may be used to smear the network output. If the matrix $S(j,i) = S(j,x(i),y(i))$ is

invertible, the network of Figure 3 can provide an interpolation (correctly reproduce the image at the sample points). If the interpolation functions are too narrow, low spatial frequencies cannot be accurately interpolated and if the functions are too broad, the accurate image reconstruction depends upon difficult-to-obtain accuracy in the inverse matrix $W(,)$.

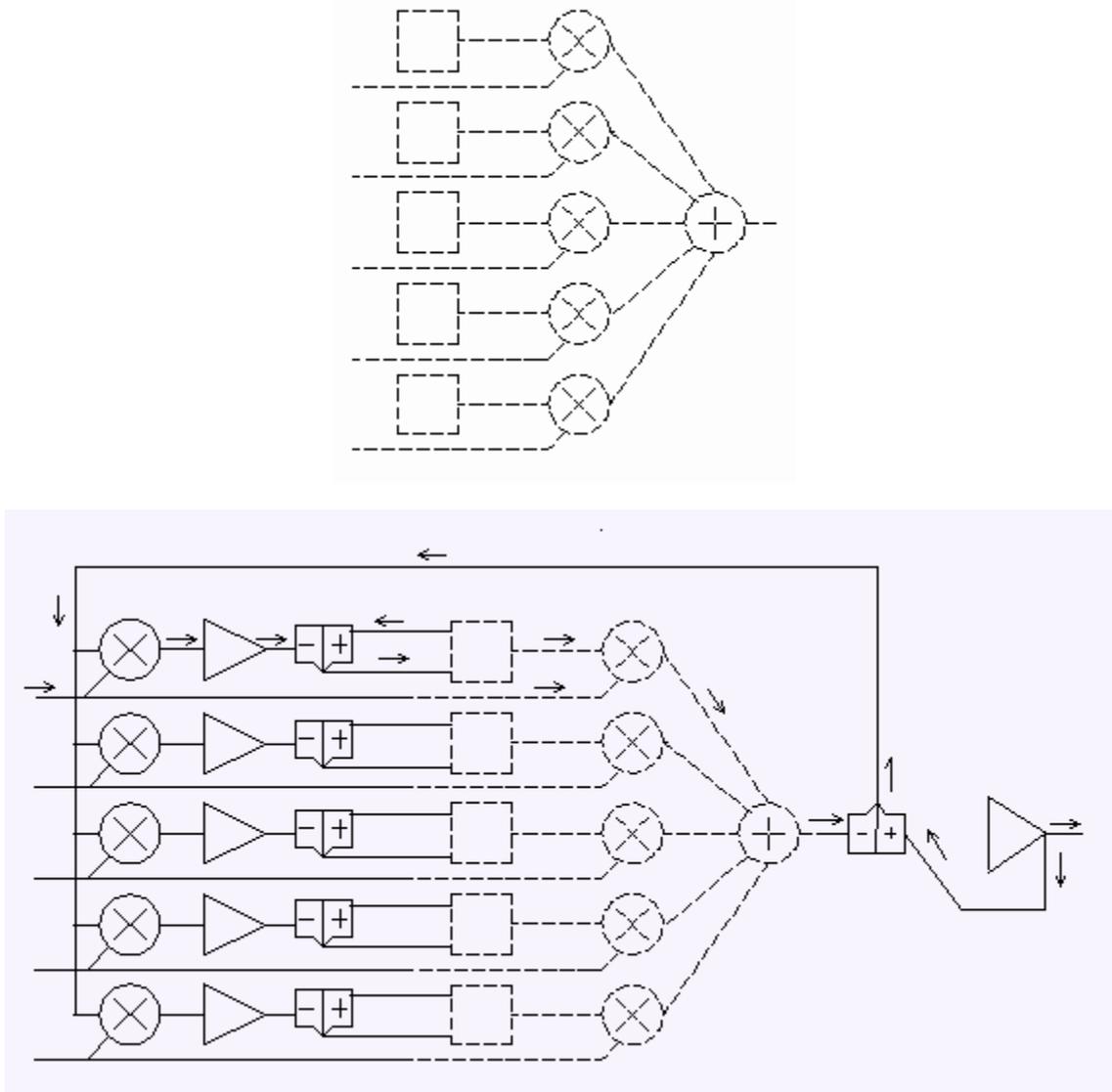
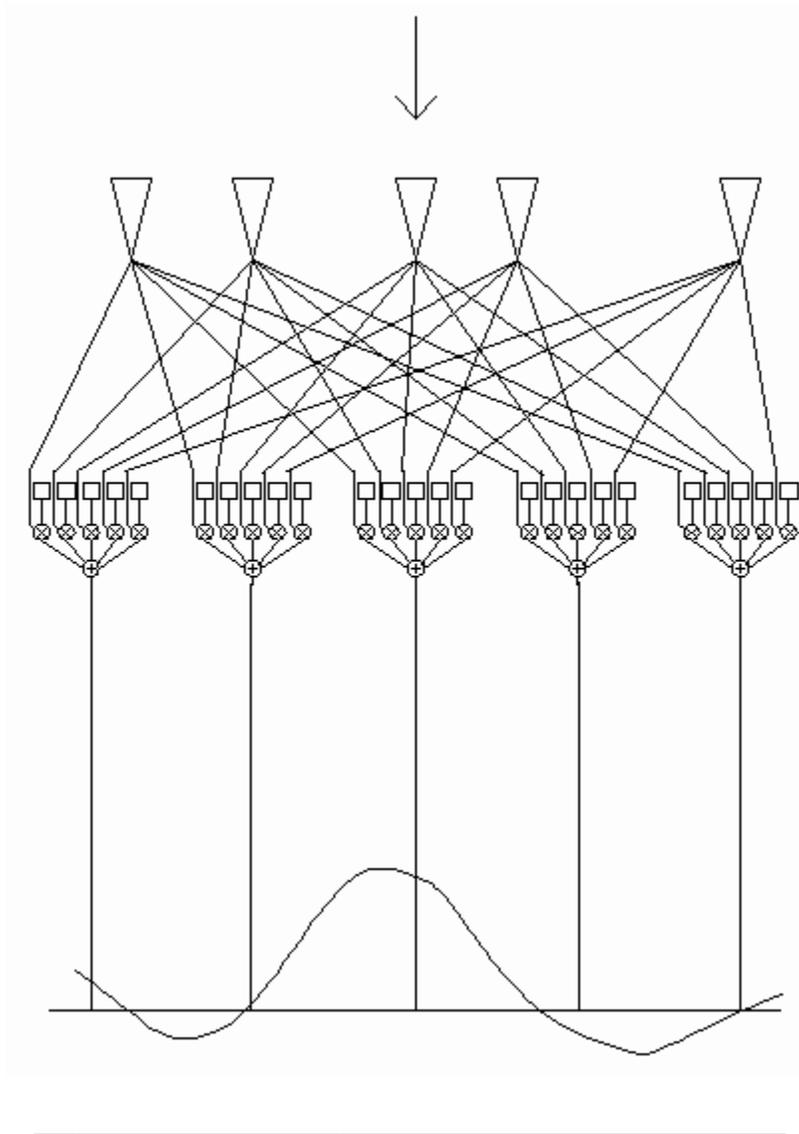


Figure 2. a - upper -) The basic "neural" network unit. It multiplies each input line by a weight and sums the products. b - lower -) The weight adjusting network for learning the weights.



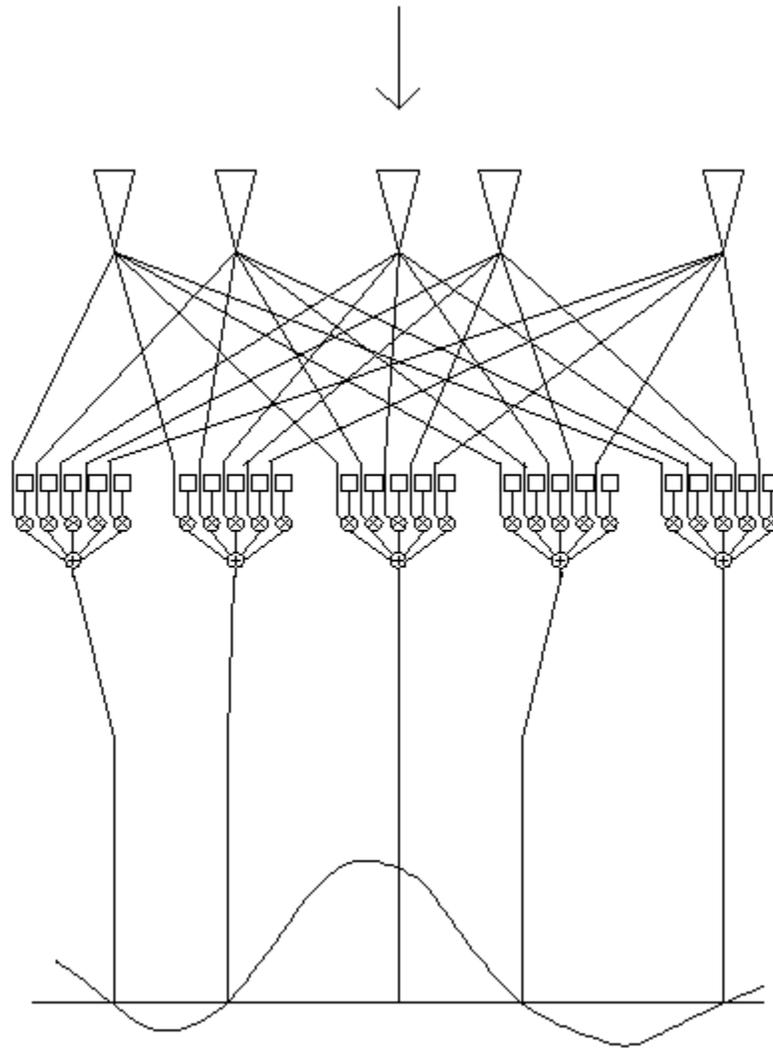


Figure 3. a - upper -) A network implementation of Yen interpolation. The weights of the network are chosen so that the weighted sum of the sinc functions generated at the output of the "neurons" is zero at the receptor positions if only one receptor is activated. b - lower -) Chen-Allenbach interpolation. Same as a) except the sinc functions are generated at the receptor positions.

To implement the interpolation scheme above, the visual system needs to compute $W(,)$. We will now describe a network implementation of a developmental process for computing these weights.

Mathematically, this will be an iterative procedure for computing the inverse of a matrix. The basic idea is to use a spread function as input to the weight matrix and then adjust the weights so that the output from the corresponding transformed unit is unity and the rest are zero, so that the final output is the input spread function. When this is done repeatedly for all the spread functions, the weight matrix will converge to the correct solution.

Figure 4 shows additional components which need to be introduced. The lines from the neural network to the spread function generation are cut and spontaneously active input units are introduced which, when active, generate the spread function for that line at output lines that carry the spread function back to the former input lines. Notice that it is only these lines which must be accurately positioned according to the retinal sampling array. Let i be the index of the spontaneously active unit, then the k th output line will have the value,

$$R(i,k) = S_j [S(i,j)W(j,k)]. \quad (12)$$

This is then subtracted from the desired output, which is provided by the new input units, to form the error term,

$$E(i,k) = I(i,k) - R(i,k), \quad (13)$$

where I is the identity transformation.

Figure 2b shows the details of the weight adjustment scheme for each unit. The error term is multiplied by the input on the old signal line, then by a gain factor, c , and then added to the previous value of the weight, giving the new weight as

$$W(j,k) = W(j,k) + cS(i,j)E(i,k). \quad (14)$$

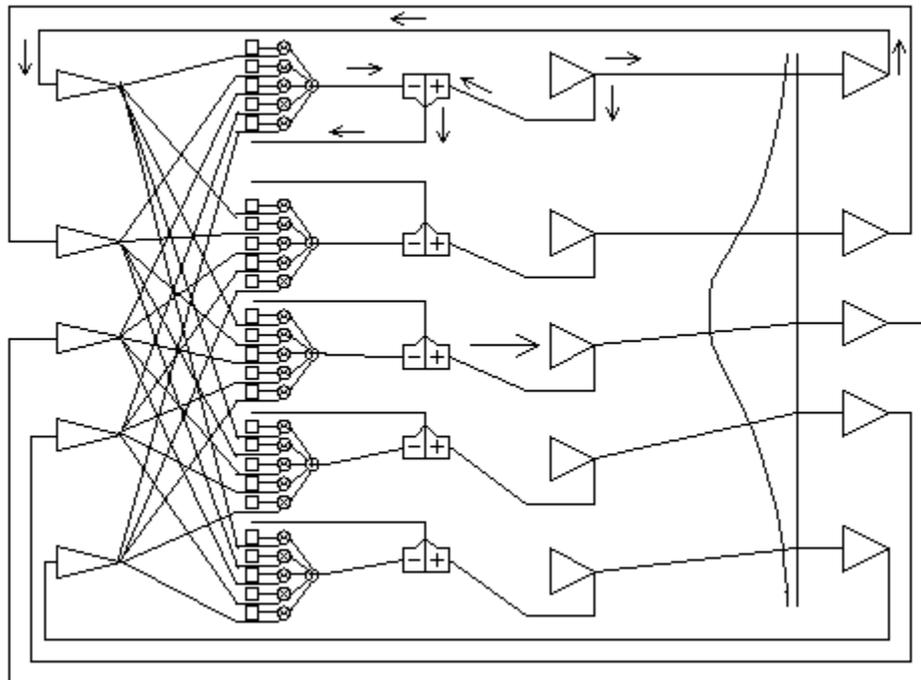


Figure 4. The learning network which feeds a spread function back into the transformation network so that the transformation weights can be adjusted. An error signal is formed by subtracting the activating pulse function from the network output. It flows back to the weight adjusting network of Figure 2b, omitted here to simplify the diagram. The arrow indicates the spontaneously active unit described in the text.

This can be thought of as an equation which is itself iterated in time until it zeros the error, or it can be seen that the errors $E(i,k)$ are reduced to zero in one step if

$$c = c(i) = 1 / S_j [(S(i,j))^2], \quad (15)$$

since then

$$S_j \{ S(i,j)(W(j,k) + c(i)S(i,j)E(i,k)) \} = I(i,k). \quad (16)$$

This shows that the multiplication by the learning rate constant c can be avoided if the sampled spread functions have unit length.

In the above equations, coefficients other than $S(i,j)$ can be used as multipliers of the error $E(i,k)$. This value

was originally chosen so that the direction of the correction would be consonant with the sign of the input for the particular weight and so that corrections would not be made to weights that had no input. Another reason for choosing this coefficient is that it is the gradient of the total squared error as a function of the $W(j,k)$. Gradient descent methods are notorious for slow convergence (13, p. 302). We have found that matrices for spread functions whose width is less than the average spacing converge rapidly, but that broad widths can lead to excruciatingly slow convergence.

By analogy to a method for improving the precision of an inverse (13, p. 42) an alternate weight adjustment is,

$$W(j,k) = W(j,k) + c(i)W(j,i)E(i,k), \quad (17)$$

which zeros the error if

$$c(i) = 1/R(i,i). \quad (18)$$

Press (et al.) (13) only discuss the behavior of the formula when the initial approximation is very close, but our experience has been that it converges much faster than the previous scheme.

A very similar formula that actually computes the inverse directly when the computation is done once for each i comes from the Sherman-Morrison method of adjusting the inverse of a matrix which is modified by adding an outer product (13, p. 66). If W is initially the identity transformation, then the adjustment for bringing in the i th weighting function is exactly

$$W(j,k) = W(j,k) + c(i)W(j,i)E'(i,k), \quad (19)$$

where

$$E'(i,k) = W(i,k) - R(i,k), \quad (20)$$

and

$$c(i) = 1/[1 - W(i,i) + R(i,i)]. \quad (21)$$

Although these latter equations are as easy for an ordinary digital computer to execute as the first procedure, they are not as easily implemented in a network. They do provide a practical way of taking advantage of the sparseness of $S(i,j)$ in calculations involving large numbers of sample points.

Discussion

Physiologists may find it interesting that receptive fields measured at the level of the output units of the interpolation network have various shapes even though they all perform the same function. This is a consequence of the variations in receptor spacing. Receptive fields of units responding to the continuous image would have additional variation corresponding to whether they are close to one output unit or midway between output units. Although units are usually specified physiologically by their receptive fields, their function in this case relates more directly to their projective fields. If this interpolation calculation were actually to be computed as described in the human visual system, it presumably would occur at the first level of cortical processing where the inputs are still not orientation specific, and there are enough cells to sample the output relatively continuously. It is possible that a calculation of this sort is "interpolated" into the processing of other features and descriptors.

Various image encoding schemes have been proposed in which an image is linearly transformed to a domain where each transform value represents a region of space and spatial frequency similar to the receptive fields of primary visual cortex orientation selective simple cells (14,15). Although some of the methods are based on orthogonal transformations which are self-invertible (16), for many of the schemes the calculation of the

inverse transformation is a problem similar to the one above: we need to invert an $n \times n$ sparse matrix where n is of the order of the number of pixels in the image. The above solutions to the inverse problem can be used in this context and the learning network can be regarded as a way to construct an encoding inversion network (17).

The learning mechanisms described above show that biologically plausible mechanisms can accomplish some useful calculations which not very many years ago were the exclusive province of large mainframe computers. The first model takes a matrix of distances as input and iteratively solves for a two-dimensional configuration of points that satisfy those distances, a problem in multidimensional scaling ordinarily solved by converting the distance matrix to a matrix of inner products and then factoring (18). The second model iteratively finds the inverse of a matrix. Both models can be thought of as models which are self-organizing in the sense that no external reinforcement is provided. However, the actual adjustment rules are those associated with error-correcting learning.

References

1. J. Hirsch and W. H. Miller, Does cone positional disorder limit resolution? *J. Opt. Soc. Amer. A*, **4**, 1481-1492 (1987).
2. [A. J. Ahumada, Jr. and A. Poirson, Cone sampling array models](#), *J. Opt. Soc. Amer. A*, **4**, 1493-1502 (1987).
3. D. J. Williams and N. J. Coletta, Cone spacing and the visual resolution limit, *J. Opt. Soc. Amer. A*, **4**, 1514-1523 (1987).
4. J. I. Yellott, Jr., Spectral consequences of photoreceptor sampling in the rhesus retina, *Science*, **221**, 383-385 (1983).
5. L. T. Maloney, Spatially irregular sampling in combination with rigid movements of the sampling array, *Invest. Ophthalmol. & Vis. Sci.*, **29**, (ARVO Suppl.), 58 (1988).
6. L. T. Maloney, Learning algorithm that calibrates a simple visual system, *Opt. Soc. Amer. Annual Meeting Tech. Digest*, **11**, 133 (1988).
7. T. Kohonen, Analysis of a simple self-organizing process, *Biol. Cyber.*, **44**, 135-140 (1982).
8. H. Ritter and K. Schulten, On the stationary state of Kohonen's self-organizing sensory mapping, *Biol. Cybern.*, **54**, 99-106 (1986).
9. J. Hirsch and R. Hylton, Quality of the primate photoreceptor lattice and limits of spatial vision. *Vision Res.*, **24**, 347-356 (1984).
10. C. L. Fales, F. O. Huck, J. A. McCormick, and S. K. Park, Wiener restoration of sampled image data: end to end analysis, *J. Opt. Soc. Amer. A*, **5**, 300-314 (1988).
11. J. L. Yen, On nonuniform sampling of bandlimited signals, *IRE Trans. Circuit Theory*, **3**, 251-257 (1956).
12. D. S. Chen and J. P. Allenbach, Analysis of error in reconstruction of two-dimensional signals from irregularly spaced samples, *IEEE Trans. of Acoust., Speech, and Sig. Process.*, **35**, 173-180 (1987).
13. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press: New York (1986).
14. E. H. Adelson, E. Simoncelli, and R. Hingorani, Orthogonal pyramid transforms for image coding, *Visual*

Communications and Image Processing II, SPIE Proc. Vol. 845 (1987).

15. A. B. Watson, Efficiency of an image code based on human vision, [J. Opt. Soc. Amer. A, 4, 2401-2417 \(1987\)](#).

16. A. B. Watson and A. J. Ahumada, Jr. An orthogonal oriented quadrature hexagonal image pyramid. [NASA Tech. Mem. 100054 \(1987\)](#).

17. J. G. Daugman, Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. *IEEE Trans. Acoust., Speech, and Sig. Process.*, **36**, 1169-1179 (1988).

18. W. S. Torgerson, *Theory and Methods of Scaling*, Wiley: New York (1958).